International Journal of Engineering Research and General Science Volume 3, Issue 5, September-October, 2015
ISSN 2091-2730

EFFICIENT ERROR CORRECTING MECHANISM FOR MEMORIES USED IN RADIATED ENVIRONMENT

Sanilkumar N S[1], Aby Thomas[2]
1.PG Scholar, Department of Electronics and communication Engineering, MACE. Kothamangalam, Kerala, India.
2. Asst.Professor, Department of Electronics and communication Engineering, MACE. Kothamangalam, Kerala, India.

Email: sanilkns@gmail.com,+91-8891769105

Abstract— Nowadays, memories we use are cheap, easily available in market, compact, have high programmability and available on various ranges of size and type. But the dis-advantage with these memories is that whenever there is a radiation strike on sensitive part of memory circuit, the data stored in the memory will be corrupted. These types of errors are common in radiation-prone environment like space, aircrafts, radiation research laboratories, power plants, mines etc. One of the remedies for this is to use error correction codes. Different error correction codes are available, from simple to complex. As the complexity increases efficiency also increases. Radiation will effect closer memory locations mainly as they are adjacent to each other. SEC-DAED code provides single error correction and double adjacent error detection, but cannot do anything about triple error detection. Codes like SEC-DED-TAED will also provide triple error detection but there is no provision for double error correction. This project compare different error correction codes and develop a new method which will provide single and double adjacent error correction which is occurring most time and also provide double non adjacent and triple adjacent error detection. The coding is done using VHDL in Xilinx 14.5 and simulations were obtained using ISim simulator. The project was implemented on Spartan3 XC3s200 FPGA platform.

Keywords—- single event effect(SEE), multiple cell upset(MCU), single error correction –double adjacent error detection (SEC-DAED),single error correction –double error –detection (SEC-DED-TAED),), Orthogonal Latin Square Codes(OLS).

1. INTRODUCTION

Reconfigurable SRAM based FPGA's are widely used in many applications due to their high programmability, high density, availability in market and cost effectiveness. Despite of these advantages they are limited use in space or air craft, because of their high susceptibility to noise or radiation; when a high energized particles striking on sensitive part of circuit there will be errors on data, which stored in that memory. If it affect only a single bit it known as single bit upset(SBU) and if it is affect more than one bits it known as multiple cell upset(MCU). Now a day's technology is scale down day by day so the radiation induced soft errors will effect more than one memory location. Normally use "error correcting" to solve these errors. Error correction codes from simple hamming codes to most complex BCH and Euclidian codes. In any error correcting codes generally encode the data when it is writing to the memory and decode when it is reading. So the encoding and decoding latency will affect the memory access time. So keep reducing the encoding and decoding latency. Single error correction codes has low latency, but if a double error occur it will miss correct; silent data corruption occurring. The radiation induced soft errors will effect closer memory locations. In many case they are adjacent to each other. This is because errors are created along the path through which it passing. This is shown in figure1.

This project make comparative study of different type of error correcting mechanism and from that produce an efficient error correcting mechanism. By this method correct all single errors, double adjacent errors and detect all non-adjacent double errors, triple adjacent errors. Different error correcting codes comparing here are single error correction-double adjacent error detection (SEC-DAED), single error correction-double error detection-triple adjacent error detection (SEC-DED-TAED), single error correction-double adjacent error correction (SEC-DAEC), single error correction-double adjacent error correction-non adjacent double error detection using OLS (SEC-DAEC-DED-OLS), and error correction using Decimal Matrix Code. Rest of this paper organized as follows. Chapter 2 analyse different error correcting codes. Chapter 3 presents proposed error correcting mechanism. Chapter 4 is the simulation results. Chapter 5 is the conclusion.

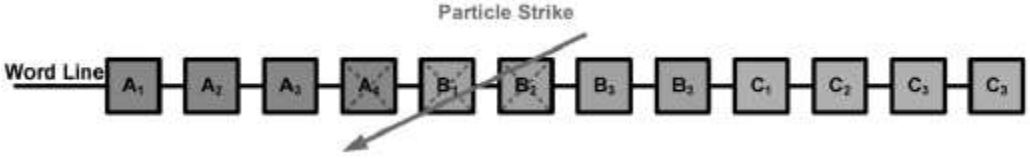

Fig 1: radiation particle strike.

2. ANALYSIS OF DIFFERENT ERROR CORRECTION MECHANISMS

The strategy using in error correcting mechanism is that encoding the data when it is writing to the memory and decoding when it is reading.

In the encoding section, from the given data bits creating extra bits, known as parity bits. If the number of information is 'k' produce 'r' number of extra bits so the total code length will be 'n'.

$$n=k + r$$

In decoding section, from the given code (n), produce another set of values known as "syndrome". Based on the syndrome determine whether there is an error or not. If the syndrome is zero there is no error else there is an error in the data. Form code word, parity bits are extracting and gives the data output. All the encoding and decoding process is based on some matrices.

Different error correcting codes comparing here are single error correction-double adjacent error detection (SEC-DAED), single error correction-double error detection-triple adjacent error detection (SEC-DED-TAED), single error correction-double adjacent error correction (SEC-DAEC), single error correction-double adjacent error correction-non adjacent double error detection using OLS (SEC-DAEC-DED-OLS), and error correction using Decimal Matrix Code.

Table 1 : Comparison table of different error correcting mechanisms.

Methods	Redundant bits	specification
Sec- daed	5	•Single error correction •Double adjacent error detection
Sec-ded-taed	6	•Single error correction •Double error detection •Triple adjacent error detection
Sec-daec	6	•Single error correction •Double adjacent error correction •Miss correct if non adjacent errors present
DMC	18	•Correction up to 5 bits •Miss correction if redundant bits have error
OLS	12	•Single error correction •Double adjacent error correction •Non adjacent double error detection •Error in redundant bit cannot correct

3. PROPOSED ERROR CORRECTION MECHANISM

The proposed mechanism is based on, SEC-DED-TAED based on hamming and OLS code. So there are two encoding modules are needed. One for hamming code and other for OLS. In the hamming encoder if we have 16 number of information data after encoding there available 22 bit of data. And this encoding based parity check matrix. And this matrix is derived from a large matrix with dimension 31 row and 26. From this matrix columns are selected with the following weight order (odd odd even odd odd even . . . and so on). . The matrix is given by:

1 0 1 **0** 1 1 **0** 1 0 1 0 1 **0** 1 0 1 0 0 1 0 1 **0**

0 1 0 **0** 1 0 **0** 1 0 0 1 0 **0** 1 0 0 1 0 0 1 0 **0**

0 0 1 **1** 1 0 **0** 0 1 1 1 0 **0** 0 1 1 1 0 0 0 1 **0**

0 0 0 **0** 0 1 **1** 1 1 1 1 0 **0** 0 0 0 0 1 1 1 1 **0**

0 0 0 **0** 0 0 **0** 0 0 0 0 1 **1** 1 1 1 1 1 1 1 1 **0**

Fig 2: matrix for sec-ded-taed code for 16 bit data word

By looking matrix, the bold columns represent parity bits and other for data bits. For find parity p1, take the first row and it is clear that positions 16,15,14,13,11,9,8,6,3,1 has value'1' so take data positions 16,15,14,13,11,9,8,6,3,1 then XOR it's values. Similarly for p2, take second row and so on. By doing this, getting 5 numbers of parity and for the last parity xor all data and newly produced parity. So total 6 parity and 16 number data so the total code length will be 22. And code word arranged in the order of parity check matrix.

$$H = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} I_{4m}$$

Fig 3: H-matrix representation of OLS code.

M_1

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

M_2

1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

M_3

1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

M_4

1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fig 4: H-matrix for OLS code.

In the case of OLS code, It has k number of data bits, where k=m2. And have 2tm extra bits. Number of errors correcting here are 'm'. OLS codes has simple and faster decoding section, it is because of decoding is based on OS-MLD. Decoding in OS-MLD based on simply taking majority value on set of recalculated syndrome values. The errors are calculated based on parity check matrix that is syndrome value is obtaining by multiplying stored code word with this parity check matrix. Parity check matrix is derived from traditional OLS matrix which given below given in fig 3.

The first k columns represent data bits and next r column for parity bits. From matrix I4m is the identity matrix of size 4m and M1, M2, M3, M4 are matrices with size m × m2. When we take only the M1 matrix its adjacent columns are sharing one, it must avoid in this proposed mechanism so avoid that matrix (M1) and also make interleave and re-arrange the matrix to avoid this type of sharing. Otherwise when there is an error in the adjacent data bits which correspond to the column values which sharing the one value will shows miss correction. The matrix reduced matrix is given below:

p_7	p_9	p_1	p_5	p_4	p_{11}	p_{12}	p_8	p_{10}	p_6	p_2	p_3
1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0
0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0
0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1
1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0
0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0
1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0

Fig.5. Reduced parity check matrix H after the removal of M1 with the proposed bit placement.

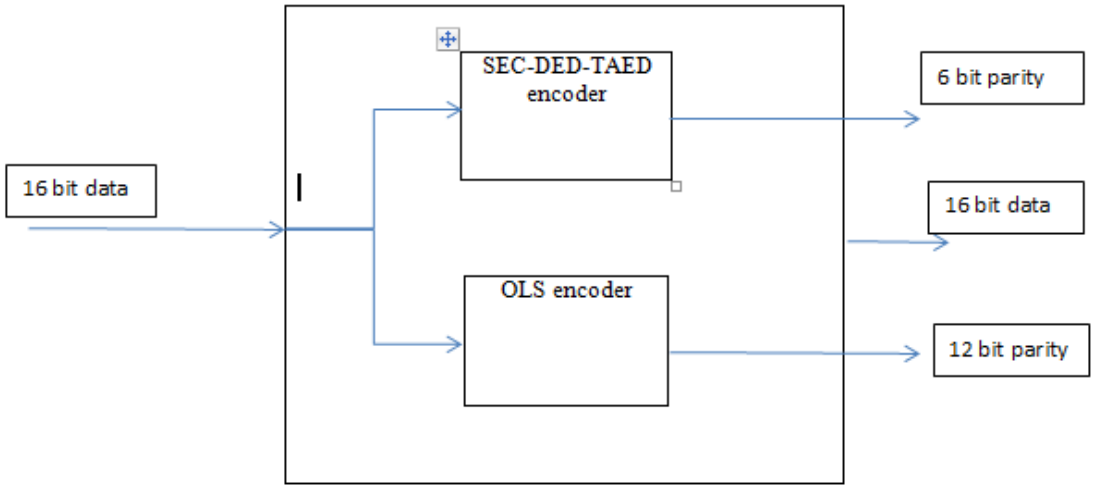

Fig 6: proposed encoder module

And the encoding is same is that of previous type, here only matrix is changed. So after encoding using OLS code there are 28 number of code word for a 16 number of data bits. So after complete encoding mechanisms of two different method there total 18 bits of parity and 16 number data and total 34 number of code word. The code word is arranged in the order given below:

P1 P2 D16 P3 D15 D14 P4 D13 D12 D11 D10 D9 P5 D8 D7 D6 D5 D4 D3 D2 D1 P6 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12

First 22 bit contains original data bits and 6 parity of SEC-DED-TAED. Next 12 bits are parity of OLS codes. The block diagram of encoder for the proposed system is given in fig 6:

In the decoding module also there are two types of decoder sections are need one for hamming and other for OLS. And in the decoding section data and parity are extracting and corresponding parity and data is passing to its corresponding decoder modules. In the case of hamming decoder, there is a parity check matrix available, “AND” the code word with each row of matrix then ”XOR” so getting a 6 bit syndrome value. To check the triple error there is one more value is need, this done by “XOR” all the code word. So after decoding there are a 6 bit syndrome and single bit parity value is available, based on these values error is determine. If syndrome is 000000 and parity is 0 then no error. If syndrome is not equal to 000000 and parity is 0 then double error. If syndrome is equal to 000000 and parity is 1 then error in last parity. If syndrome is correspond to any column value and parity is 1 then single error in data bit. If syndrome is correspond to any other value and parity is 1 then triple error in data bit.

In the case of OLS code, there is also a parity check matrix available. So based on this matrix find out the syndrome value, here syndrome is 16 bit. Based on this syndrome error is determine. Error correction based on the figure 7.

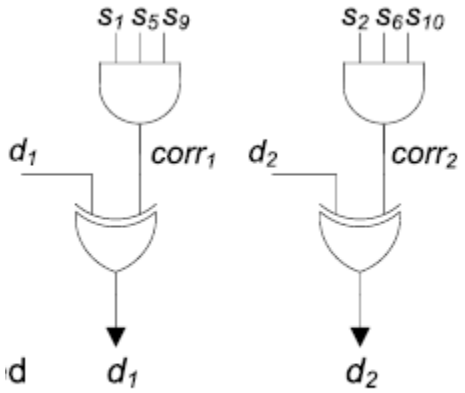

Fig 7: Illustration of the decoder for data bits 1 and 2.

Diagram show error correcting circuit for the first two data bits, where s_i is the corresponding syndrome value of data. There is also a provision to detect the double non adjacent error, which is based on figure 8.

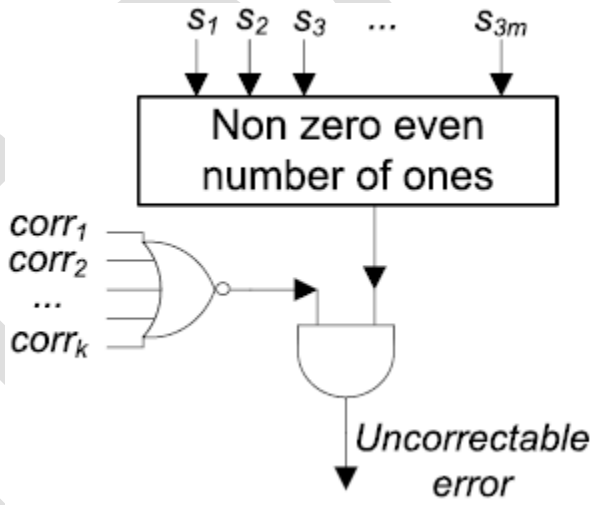

Fig.8. Detection of double un correctable errors in the proposed scheme.

It is based on the fact that when there is a double error, the syndrome value is become even value. If there is a double adjacent error is it can correct using previous diagram but when there is non-adjacent error is present it cannot correct. So if there is a nonadjacent double error is present, it indicated as un correctable error.

The out puts of hamming code are 16 bit data, single_error1 for single error, double-error1 for double error, taed1 for triple adjacent error and parity_error1 for last parity. The out puts of OLS code are 16 bit data and "ue" for indicating un correctable errors. The proposed error correcting mechanism is based on the proper selection from these out puts.

The selection is based on the following strategy:

If there is no errors indicating in both mechanism, then there is no error happened to data then select output from any mechanism. If there is only a single error in SEC-DED-TAED code and no error for OLS code, it is corresponds to a parity error in SEC-DED-TAED code and no error to the data, it can be corrected by either method. So take output from any mechanism and tagged it as no error. If there is a double error occurs in SEC-DED-TAED code and no error to OLS code, then it is a double error for parity only in the SEC-

DED-TAED code. So select the outputs of OLS code. If there is double error in SEC-DED-TAED code and un correctable error in OLS code it correspond to a non-adjacent error. It cannot be corrected it only detect. If there is triple adjacent error in SEC-DED-TAED code and un correctable error in OLS code it correspond to a adjacent triple error. It cannot be corrected it only detect. The block diagram for the decoder of proposed mechanism is given below

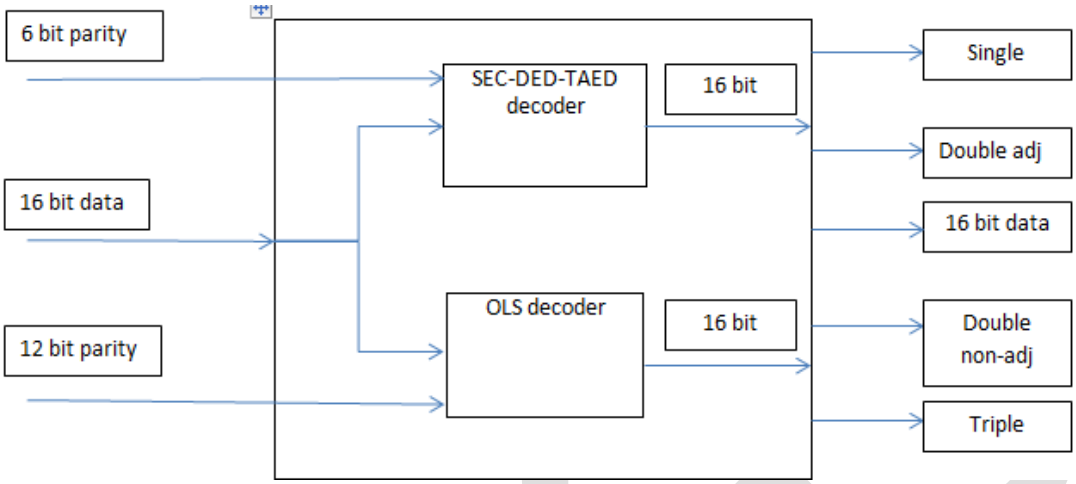

Fig 9: proposed decoder module

4. SIMULATION RESULTS

If there are 16 number of data after encoding there 34 number of code words. Result of encoder module given below:

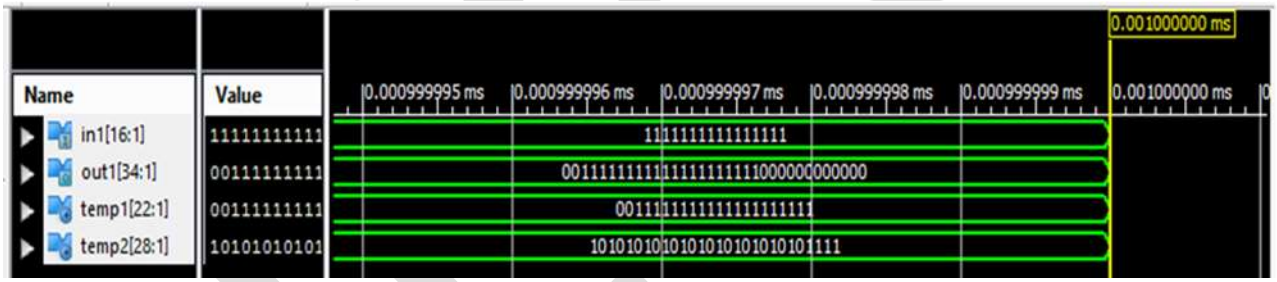

Fig.10. Result of encoder

If there is a single error, error in the third position of code word, correct the value and extract the data bits from code word and error indicators points to zero

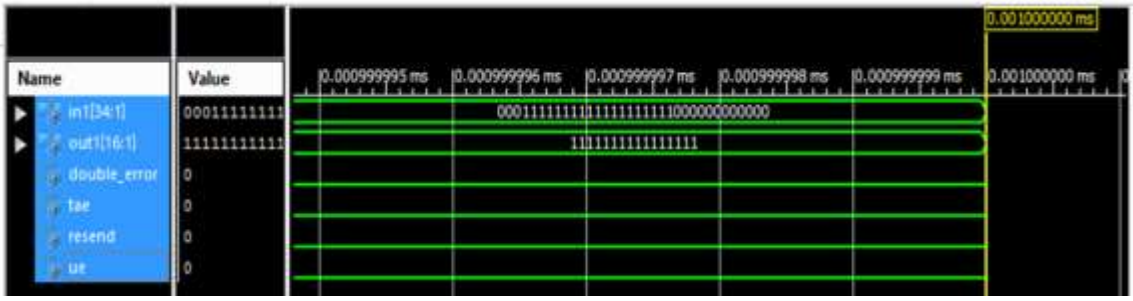

Fig.11. Result of decoder with single error

If there is a double adjacent error, error in fifth and sixth position of code word, correct the value and extract the data bits from code word and error indicators points to zero

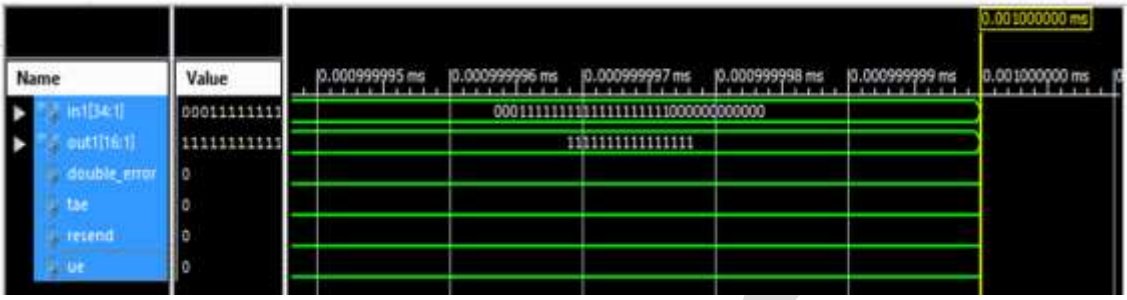

Fig.12. Result of decoder with double adjacent error

If there is a non-adjacent double error, error in third and nineteenth position of code word, cannot correct the its value and extracted the data bits from code word are un known but error indicators, that double-error and ue is points to one. Also there is an indicator that is resend, it is also high.

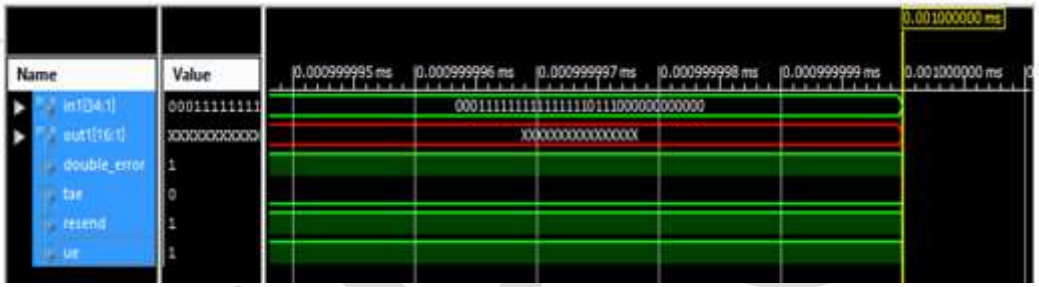

Fig.13. Result of decoder with double non-adjacent error

If there is an adjacent triple error, error in eighteenth, nineteenth and twentieth position of code word, cannot correct its value and extracted the data bits from code word are un known but error indicators, that is tae and ue points to one. Also there is an indicator that is resend, it is also high.

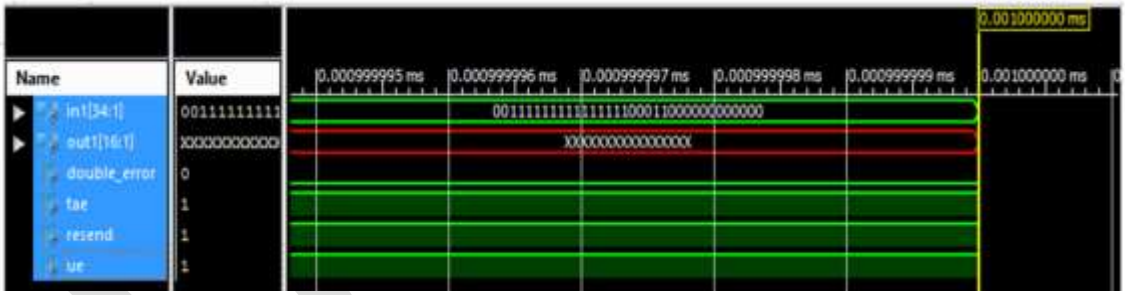

Fig.14. Result of decoder with triple adjacent error

5. CONCLUSION

There are different error correcting mechanisms are available, by compare and analyses them properly it is understand that each has its own advantages and limitation. The proposed mechanism which derived from the existing codes, single error correction-double error-detection –triple-adjacent error detection using hamming code and single error correction-double error detection –double adjacent error correction Codes Derived From Orthogonal Latin Square Codes will provide single and double adjacent error correction and double nonadjacent and triple error detection. These all combination is not provided any other codes. But here the number of parity bits increased compare to others so this code will use precision is greater than cost. The code is implemented in Spartan 3 FPGA trainer board. This project provides correction up to 2 errors that is single and adjacent double errors are corrected. And detection up to 3 errors that is detecting double non adjacent errors and triple adjacent errors. Radiation will affect closed memory locations mainly they are adjacent locations. So there is higher chance to occur double adjacent errors, triple adjacent errors etc. But this project provides correction up to double adjacent errors so the future works are to correct triple errors.

REFERENCES:

[1] S. Satoh, Y. Tosaka, and S. A. Wender, "Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's," IEEE Electron Device Lett., vol. 21, no. 6, pp. 310–312, Jun. 2000.
[2] R. W. Hamming, "Error detecting and error correcting codes," Bell Syst.Tech. J., vol. 26, no. 2, pp. 147–160, Apr. 1950.
[3] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 305–316, Sep. 2005.
[4] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
[5] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop.,vol. 28, no. 2, pp. 124–134, Mar. 1984.
[6] H. Lin, F. Yeh, I. Chen, and S. Kuo, "An efficient perfect algorithm for memory repair problems," in Proc. 19th IEEE Int. Symp. Defect Fault Toler. VLSI Syst., Oct. 10–13, 2004, pp. 306–313.
[7] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
[8] P. Reviriego, M. Flanagan, and J. A. Maestro, "A (64,45) triple error correction code for memory applications," IEEE Trans. Device Mater. Rel., vol. 12, no. 1, pp. 101–106, Mar. 2012.
[9] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.
[10] C. Argyrides, D. K. Pradhan, and T. Kocak, "Matrix codes for reliable and cost efficient memory chips," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 3, pp. 420–428, Mar. 2011.
[11] M. Richter, K. Oberlaender, and M. Goessel, "New linear SEC-DED codes with reduced triple bit error miscorrection probability," in Proc. 14th IEEE IOLTS, Jul. 2008, pp. 37–42.
[12] C. A. Argyrides, P. Reviriego, D. K. Pradhan, and J. A. Maestro, "Matrix-based codes for adjacent error correction," IEEE Trans. Nucl. Sci., vol. 57, no. 4, pp. 2106–2111, Aug. 2010